

Teleoperation for a Ballcatching Task with Significant Dynamics

Christian Smith*, Mattias Bratt, and Henrik I Christensen

(Invited Paper - Technology and Applications)

Abstract—In this paper we present ongoing work on how to incorporate human motion models into the design of a high performance teleoperation platform. A short description of human motion models used for ballcatching is followed by a more detailed study of a teleoperation platform on which to conduct experiments. Also, a pilot study using minimum jerk theory to explain user input behavior in teleoperated catching is presented.

Index Terms—Teleoperation, control, high performance manipulation, human motion.

I. INTRODUCTION

The desire to transfer human action over space, time and scale is well-known by now. One of the most widely advertised examples is the NASA Mars Rover Program with Sojourner, Spirit and Opportunity. Other use of teleoperation includes medical robotics and recent teleoperated drones for military surveillance. Teleoperation received its initial popularity during World War II for handling of nuclear material.

Early work emphasized transfer across space to allow people to manipulate dangerous material (Goertz, Blomgren, Grimson, W. M. Thompson, & Kline, 1961). The degree of autonomy was non-existing and the systems were designed for direct motion reproduction. The systems were often limited by coarse feedback and limited visibility of the workspace.

Recent work such as operation of the MARS rovers have transferred actions across time, due to the significant time-delay of operating vehicles on another planet (Tsumaki et al., 2000). The degree of autonomy is significant to enable efficient operation and to avoid direct closure of the control loop, which would be close to impossible.

Teleoperation involves two closely related problems, control of a system in the presence of transfer function that is non-linear, which could be transfer across space (for example changes in kinematics), transfer in time (delays that are significant compared to the time constant of the target system), and transfers across scale (significant changes in scale for example for nano-robotics or as part of medical surgery). The second problem is generation of feedback to the user to give him/her the best possible task embedding, or sense of presence. The feedback requirements are also closely coupled to the degree of autonomy in the system.

Few studies have considered teleoperation in the presence of significant dynamics (Hirzinger, Landzettel, & Fagerer, 1994; M. R. Stein & Paul, 1994). The qualifier “significant” dynamics is here used to specify a process where the time-delay is a large fraction of the process cycle time. Such processes could for example be control of aerial vehicles.

For processes with significant dynamics it is of interest to study the influence of varying degrees of autonomy and to be able to understand the limits as to how a user can be embedded in a task context, that is, how the user can be given a sense of presence for actual interaction with objects and how can this be achieved in the presence of delays so as to find a suitable balance between delegation/tasking and actual control from the user. It is also of interest to study different degrees of embedding including the influence of use of different feedback modalities such as haptics and visual feedback.

To understand this problem we have chosen to study the challenge of teleoperated ball catching. For catching of balls thrown across a room the flying time is on the order of 0.5-1.0 seconds, and for teleoperation across the internet typical delays are in the range 0.1-0.4 seconds for normal socket operations. The autonomous catching of balls using a robot has for example been studied by DLR (Frese et al., 2001), Andersson (Andersson, 1989), Hove (Hove & Slotine, 1991), and others.

Teleoperation has been widely used in a variety of applications and is in many respects considered a mature technology as described in Section II. The design of teleoperation systems for such applications are required to carefully consider the integration of different sensory modalities, which motivates studies of human perception-action models, as outlined in Section III. It is well known that humans are visually dominated (Schwartz, Moran, & Reina, 2004) and it is thus of interest to understand how this influences integration of haptics and graphics. At the same time there are clear models that suggest that much of human actions are feed-forward driven rather than feedback driven (Berthoz, 2000). Under such a control regime one would expect that motion for interaction is largely prospective, in particular for dynamic situations. The control strategy for human catching has been studied for some limited cases, as described in Section III-C. However, general human ball catching strategies are poorly documented in the literature.

To study the overall problem of ball catching we have designed a high performance robotic manipulator. An objective of the design has been use of components off the shelf, to make it possible for others to replicate the setup with a minimum of effort. The platform design is discussed in

The authors are with the Center for Autonomous Systems, Swedish Royal Institute of Technology, Stockholm, Sweden <http://www.cas.kth.se>

e-mail addresses: {ccs,bratt,hic}@kth.se

*Correspondance to Christian Smith, Center for Autonomous Systems, Kungliga Tekniska Högskolan, S-100 44, Stockholm, Sweden

phone: +46 8 790 6203, fax: +46 8 723 0302

Manuscript received March 1, 2007

Section IV. For operator feedback both regular graphics and stereoscopic displays have been studied in combination with force-feedback joysticks. The design of a range of different interaction modalities is also documented.

The system has been implemented and a structured set of experiments have been designed. The experimental design is outlined in Section V. The initial results obtained with the system are documented in Section VI and the general observations / lessons are presented in Section VIII.

II. TELEOPERATION CONTEXT

Teleoperation was originally introduced during World War II (Goertz, 1952; Whitney, 1987). The systems were of the master-slave kind. A human user was operating a master input device and the slave manipulator reproduced the exact motion. Often the master and slave were coupled mechanically. With a gradual introduction of electronic control systems and modern control theory the mechanical link was replaced by transfer through communication. With the communication over an electronic media there is a delay between operator instructions and slave motion; the same is also true for generation of feedback to the user (Hirzinger et al., 1999). The introduced delay challenges traditional control models that assume signals to be delivered instantly. A frequently used model is to introduce a Smith predictor for compensation for the system delay (Åström & Wittenmark, 1995; O. J. M. Smith, 1959). The Smith predictor inherently limits bandwidth of the overall system and in addition it is only well suited for systems with deterministic delays. Others have considered systems with varying time-delays for handling stochastic channels (Munir & Book, 2002). The idea is here to explicitly include a model of the communication channel in the systems model. These systems have good performance for remote haptic manipulation etc. However, the bandwidth of interaction is still limited.

For applications with significant time-delays, such as the Mars rover program or control of systems on-board the international space station or the space shuttle, the adopted strategy has been to use task oriented control (Muirhead, 2004; Hirzinger et al., 1994; M. R. Stein & Paul, 1994). Rather than trying to perform direct tele-operation, the operator specifies the overall strategy and sequencing of tasks. The detailed feedback control is delegated to the robot system. The challenge is here to have continuous control and feedback from the system.

III. ISSUES DERIVED FROM HUMAN STUDIES

There are several issues that emerge in the study of human-machine interaction and human ball catching that influence human operated robotic ball-catching. This section gives a brief overview of the studies that are relevant to the present work, along with the implications they have for the teleoperation scenario.

A. Visual Dominance over Proprioception

Humans have access to an array of different senses using different modalities. It is a common view in recent neurophysiological research that these differences in modality are

used not only to register different types of real phenomena, but also to register different aspects of the same phenomena. For example, when handling an object, it can be perceived by vision as well as touch. Indeed, this multisensory processing seems to be the rule rather than the exception, and there is evidence that the integration of sensory signals takes place at a very basic level of the brain's sensory processing (B. E. Stein & Meredith, 1993; Shimojo & Shams, 2001; Schwartz et al., 2004).

In natural settings, it would be expected that spatially and temporally coinciding stimuli of different modalities would originate from the same phenomenon, and similarly, that stimuli that originate from the same phenomenon would coincide in time and space. However, when stimuli are artificially generated — e.g. in a teleoperation interface — there is a possibility to produce completely independent stimuli for different modalities. Experiments have shown that when presented with objects that have visual shapes that differ from their tactile shapes, most subjects do not even notice the discrepancy (Rock & Victor, 1964; Shimojo & Shams, 2001). Both studies show that not only can the objects be scaled differently, but they can also be significantly deformed between the modalities without the subjects reporting any discrepancies. The same has been shown for discrepancy in stiffness in a haptic device and visual cues (Srinivasan, Beauregard, & Brock, 1996).

It is also noteworthy that all three studies indicate that when presented with discrepant haptic and visual information, most subjects tend to accept the visual cues as the “true” version, and report that the tactile information is identical to this. This is especially true for the experiments on visual versus proprioceptive modalities (Shimojo & Shams, 2001). Here, subjects moved their hands in elliptical trajectories where the proportions of the major and minor axis were as large as 2:1, and would report that they were moving their hands in perfect circles if presented with a visual stimuli of a perfect circle trajectory that otherwise coincided with their hand movement. It was even possible to change the mapping from hand motion to visual motion mid-experiment without the subjects noticing, as long as the visual stimuli stayed unchanged.

This suggests that when designing a multimodal user interface, it is not necessary for haptic and visual information to coincide perfectly, and that it could be possible to use different scales and non-isotropic gains in mappings between modalities. This is interesting when the physical workspace of the operator does not match that of the manipulator, since a rescaled non-isotropic mapping could potentially bridge kinematic differences between operator and manipulator without violating the operator's sense of presence.

B. Feedback vs. Ballistic Control

Another subject that is relevant for teleoperated ball-catching is the study of ordinary human ball-catching as it is performed in various sports and games. The possible strategies for ball-catching can be divided into two categories, *predictive* strategies relying on ballistic models, or *prospective* strategies utilizing feedback (Dessing, Peper, Bullock, & Beek, 2005).

1) *Predictive Ball-catching*: Predictive ball-catching relies on a ballistic model of ball motion and observed initial conditions to make a prediction of the most probable trajectory. The differential equation governing a ballistic trajectory is presented in Equation 1, c.f. (Frese et al., 2001).

$$\ddot{X} = -\dot{X} \cdot |\dot{X}| \alpha - \vec{g} \quad (1)$$

Here X is the (3D) position of the object, dots indicate the time derivative, α is the coefficient of drag, and \vec{g} is the gravity vector. In a normal windless Earth environment, these factors are necessary and sufficient to correctly predict the motion of a thrown ball.

A predictive strategy is necessary when the motion commands have to be started well before the point of impact, i.e. when the time of flight of the ball is short compared to the reaction and/or motion time of the catcher. This is typical for batting in baseball or cricket (Dessing et al., 2005; Land & McLeod, 2000).

2) *Prospective Ball-catching*: Prospective strategies utilize continuous feedback from the task performed. There is no explicit need for an exact model of target motion, but instead, motions are made to minimize the distance or relative velocity between object and catcher. This type of strategy is viable when there is enough time to perform several corrections before the point of impact, and is useful when the trajectory is difficult to predict from initial conditions. This is typical for outfield players in baseball (Dessing et al., 2005).

3) *Combined Strategies*: When catching a ball that is thrown across a room at a moderate velocity of approximately 5 to 6 m/s, The time for the entire throw is approximately 0.8 s. In one study, it was found that the average hand motion time for a human subject attempting to catch such a ball was 0.52 s, and that the average hand trajectory length was approximately 0.35 m (Krüger, 2006). In these cases, there does not seem to be enough time for a conscious continuous feed-back loop to correct the hand position, but rather an initial estimate is made of the ball trajectory, and the hand moved roughly toward a point of intercept. If there is still time to react, this position can be corrected one or more times to produce an accurate catch (Hauck, Sorg, & Schenk, 1999). Evidence towards this is that one can distinguish a distinct trajectory towards an initial catch position estimate, and additional shorter distinct trajectories that correct this.

C. Human Ball Catching

As mentioned above, the task of a baseball outfielder has been studied in the context of prospective catching strategies. It can be viewed as an extreme case of catching behavior because of the long flight time — several seconds — of the ball and the large distance that the fielder has to cover to the interception point (Dessing et al., 2005). The task is in essence two-dimensional, since the ball can only be caught when it has (almost) reached the ground. Furthermore, the viewpoint can be considered to be collocated with the hand, except maybe at the very last stage of catching.

These properties make baseball outfielder catching behavior a good candidate for modeling by prospective feed-back

control, relying solely on 2D retinal information to guide the player towards the goal point, where the ball will reach the ground. Indeed, there are several attempts in literature to determine such a control law from data collected from outfielder catching. The *Linear Optic Trajectory* (LOT) model was proposed by McBeath *et al.* in (McBeath, Shaffer, & Kaiser, 1995), but has been opposed by McLeod *et al.* (McLeod, Reed, & Dienes, 2002), who instead advocate the *Optic Acceleration Cancellation* theory. It seems that this controversy might stem from the fact that it is difficult to determine what optic information based control law, if any, a fielder uses just from observing his trajectory and that of the ball during catching. As explained in (Marken, 2005) it is also necessary to introduce disturbances into the hypothesized controlled variable and observe the extent to which these have the expected effect, or if the variable is protected from the disturbance by control.

Catching at smaller distances, where the viewpoint location is clearly separate from the position of the hand, has been studied in terms of the internal models that humans might apply to the catching task. Specifically, McIntyre *et al.* have shown that a priori knowledge about gravity direction and magnitude is used when catching falling balls (McIntyre, Berthoz, Zago, & Lacquaniti, 2001). Catching actions taken by astronaut subjects were timed relative to the impact of the ball, on the Earth as well as under micro-gravity conditions in orbit. Results show that subjects anticipate impact at a point in time before it actually happens when performing the experiment in micro-gravity. This demonstrates that they are not able to measure the acceleration of the ball in real-time. The magnitude of the time shift when changing the initial velocity of the ball also cannot be explained by the brain using a first order, constant velocity model, even if corrected by scaling. However, good agreement is found with an internal second order model of gravity. In 0 g, the down direction of the internal model is taken from visual cues, such as walls, ceiling with lighting, etc.

Concerning hand trajectories for catching and other reaching tasks, several important characteristics can be noted. Trajectories are approximately straight in Cartesian space, and the magnitude of the velocity has a bell-shaped profile (Kawato, 1996). More complex trajectories seem to be well described by superpositions of several simple ones, as proposed in (Morasso & Mussa Ivaldi, 1982) and further investigated in (Milner, 1992). Similar bell-shaped velocity profiles can be synthesized using different optimization criteria, such as minimizing jerk (the time derivative of acceleration), joint torque change, energy, or force (Kawato, 1996; Hauck et al., 1999).

Recently, studies at the Deutsches Zentrum für Luft- und Raumfahrt (DLR), have shown similar results for human 3D catching (Krüger, 2006). Specifically, there was almost always a local minimum in the velocity profile before the catching instant, termed the *point of minimum velocity* (PMV). The consistent existence of a PMV strongly suggests that there is a first gross motion towards the catching point, followed by one or more corrective submotions as described in Section III-B3. For subjects with much experience of ball catching from sports, submotions after the PMV also acted to reduce the impact of the ball.

D. Trajectory Optimization

A well-known model for explaining the kinematics of human visually guided reaching motions is known as the *minimum jerk* (MJ) model, and was briefly touched upon in Section III-C above. It was first proposed by Hogan for single-joint motions (Hogan, 1985), and later extended to include multi-joint planar motion (Flash & Hogan, 1985).

The key observation of Hogan *et al.* was that the trajectory of voluntary arm motions, when described in extra-corporeal cartesian space, followed certain constraints. It seemed that the trajectories could be predicted by using a model in which the square sum of the third derivative of position, *jerk*, integrated over time was minimized, hence the name. In other words, given a starting point, an end point and a time to move between the two, the trajectory that minimizes the jerk on this interval is the MJ trajectory.

The MJ solution was also found for the general case of movement. It was shown that all MJ trajectories share the property that the 6th derivative is zero for the duration of the motion, and that they thus can be described as 5th degree polynomials, as in Equation 2.

$$\mathbf{x}(t) = a_1 t^5 + a_2 t^4 + a_3 t^3 + a_4 t^2 + a_5 t + a_6 \quad (2)$$

If we also add the start and end points of the motion, $\mathbf{x}(t_0)$ and $\mathbf{x}(t_1)$, and state the position, velocity, and acceleration at these points, we get the following constraints on Equation 2.

$$\begin{aligned} \mathbf{x}(t_0) &= \mathbf{x}_0, & \mathbf{x}(t_1) &= \mathbf{x}_1 \\ \dot{\mathbf{x}}(t_0) &= \dot{\mathbf{x}}_0, & \dot{\mathbf{x}}(t_1) &= \dot{\mathbf{x}}_1 \\ \ddot{\mathbf{x}}(t_0) &= \ddot{\mathbf{x}}_0, & \ddot{\mathbf{x}}(t_1) &= \ddot{\mathbf{x}}_1 \end{aligned}$$

The above constraints will give us 6 equations, and we get a well-defined system to find the 6 parameters $a_1 \dots a_6$. Thus, there is only one possible MJ trajectory for a given start and end, and it can be found by solving a simple system of linear equations. This is well-studied, and thorough descriptions and analysis for generating MJ paths for different situations can also be found in the literature (Kyriakopoulos & Saridis, 1988; Piazza & Visioli, 2000; Secco & Magenes, 2004).

The trajectories described by the MJ model are of course limited to one single motion. What happens if a more complex motion is desired or if the target of the motion is changed in mid-motion can be described by superpositioning several MJ trajectories. It is obvious that if the added MJ trajectory has an initial position, velocity, and acceleration of zero, this will still result in a continuous motion where the 6th derivative is zero, so the jerk is still minimized. This has been thoroughly described (Flash & Henis, 1991; Gat-Falik & Flash, 1999). The interesting point is that with this description, even when the target position is changed, the original MJ motion is not removed from the trajectory, but kept alongside the new motion. A new submovement can be generated as often as once every 100 ms (Milner, 1992).

For completeness, it should be noted that other models also have been proposed to explain the kinematics of human motor trajectories. They differ mostly in what cost function the target trajectory minimizes. Nelson gives a compilation

of trajectories given by different target functions (Nelson, 1983). Some of these target functions, like *time*, *peak velocity* and *peak acceleration*, can be easily discarded as not fitting well with observed data for reaching motions. Other target functions, like *energy* and *impulse* need detailed models of the dynamics of the arm, such as link impedance, frictions, and muscle forces and are therefore difficult to implement without detailed physiological measurements of the subject.

The same applies to the minimum torque change model (Kawato, 1996). This also takes into account the dynamics of the arm, and strives to minimize the produced joint torque. The trajectories described by this model produce a better fit to observed human motion data than MJ trajectories when the motion is perturbed by an outer force, or includes significant changes in posture. However, the minimum torque change model does not produce better fits to measured data than the MJ model in the absence of external forces or major posture changes.

Based on observation that the noise in the human motor system is proportional to the signal amplitude, a model where the trajectory of a reaching motion is explained by minimizing the variance of the final position has been suggested (Harris & Wolpert, 1998). While this model may have more biological relevance as to explaining why a certain trajectory is chosen, the trajectories it predicts for reaching motions do not differ significantly from those given by MJ, and thus does not add to this application.

IV. PLATFORM FOR EMPIRICAL STUDIES

A. Control Strategy for Teleoperation

Direct teleoperation has a number of different challenges depending on the problem at hand, including kinematic transfer, handling of time delays, and feedback generation. Kinematic transfer refers to the fact that in some cases there is a difference in kinematic structure between the actuation system (the manipulator) and the control system available to the operator. In control theory the handling of deterministic delays is often modeled using a Smith predictor structure, whereas there are a number of different strategies for handling of stochastic delays such as (Munir & Book, 2002). Finally the issue of feedback generation is important as multiple modalities often can be used to complement each other and provide efficient embedding of the operator in the scenario.

We use stereoscopic VR modeling and a force-reflectance joystick to provide flexibility in the feedback generation and best performance in the type of experiments to be performed. The study is performed in the context of control of a system over the Internet, so the system is assumed to have stochastic time characteristics. A common technique for closing a teleoperation control loop over a communication link with significant delays is to use wave variables (Niemeyer & Slotine, 1997). Whereas this can guarantee passivity (no net output of energy) of the communication link, and thereby help prove stability of the total system, it imposes severe performance penalties in a scenario where fast dynamics are required despite the presence of substantial communication delays.

For this reason a different approach, requiring an accurate dynamic model of the robot and its environment, was selected.

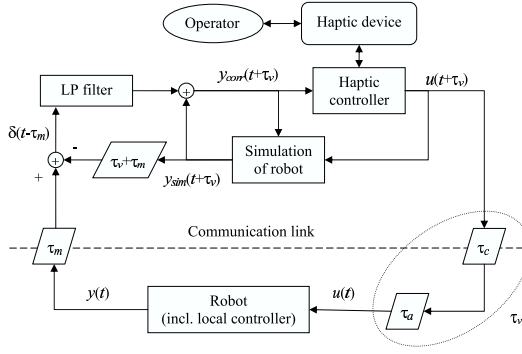


Fig. 1. Control structure (skewed boxes represent delays)

In the employed scheme, all forces displayed to the operator through the haptic device are generated interacting exclusively with a dynamic simulation running on the operator interface computer. Since this interaction does not involve a time delay, the risk of instabilities is much reduced. The modeling of the robot is facilitated by including the robot's local controller in the model. Knowing the input command and the dynamic state of the robot, the future state can be predicted well after a time of the same magnitude as the communication delays.

To close the control loop over the robot via the communication link, we use a non-linear, multivariate Smith predictor control structure (depicted in Figure 1). Because the system simulated (the robot and its controller at the robot site) in the predictor is highly non-linear, it is not sufficient to correct only the predictor output by the measured state y (the joint angles and speeds) arriving from the robot, as in a standard Smith predictor (O. J. M. Smith, 1959). The simulation itself also needs access to the corrected state estimate to be able to use the correct dynamics for that particular point in state space. The simulation output however, is not corrected, and can still be compared to incoming measurements for generating new corrections.

The command and measurement communication delays, τ_c and τ_m respectively, are handled by

- i) Adding an artificial delay τ_a to the stochastic command delay τ_c when a command packet arrives at the robot so that their sum, the virtual delay τ_v , is constant (Kosuge, Murayama, & Takeo, 1996).
- ii) Delaying the simulation result $y_{sim}(t + \tau_v)$ by $\tau_v + \tau_m$ before it is compared to the measured state $y(t - \tau_m)$ to form the simulation correction δ . This means that when a measurement packet arrives from the robot, the current value of τ_m is calculated from timestamps and an old simulation result $y_{sim}(t - \tau_m)$ retrieved from memory for comparison.

The net effect of this compensation is that the simulation, the haptic controller, and the operator all perceive time τ_v ahead of real time. That allows the generated command signal u to travel to the robot before it is needed by the robot controller. Values of τ_v upto 200 ms have been tested successfully.

Because the correction loop closed over the communication

link runs at a much lower frequency (~ 50 Hz) than the inner loop containing the simulation and control of the haptic device (~ 500 Hz), the correction signal δ must be low pass filtered to reduce zero order hold noise that would otherwise be felt by the operator as vibrations. The low pass filter also reduces the high frequency gain of the correction loop, which moderates the spike effect of quick state changes in the simulation and at the robot being slightly offset in time.

The basic principle of the haptic controller of Figure 1 is to map the haptic device handle to the chosen command space (position in the case of teleoperating the robot arm) and virtually attach it by a spring to the current corrected simulated robot position. I.e., the current 3D position of the end-effector is mapped to a point in the haptic device workspace to which the handle is attracted by a virtual spring. The command sent to the robot is just the position of the handle translated to an end-effector position by the inverse of the same mapping. In this way the force perceived by the operator is related to the resistance of the robot and its controller to adjust to the command.

In addition, virtual viscous damping is applied to the handle to help stabilize the system, and in the case of velocity control, a weak force toward zero velocity is added as well. To reflect the quality of the communication link, both of these forces can be strengthened in response to any sustained increase in the measurement communication delays.

To avoid known obstacles, these are mapped into the used command space, and made to repel the handle with a force proportional to the inverse of the square or cube of the distance to the handle depending on the dimensionality of the space. For position control this mapping is trivial, but mapping obstacles to velocity space requires some thought.

The mapping chosen for velocity control is based on the principle that a point in velocity space is part of an obstacle if, starting at this velocity v , and at the current position of the robot, braking with a postulated constant negative acceleration a_{brake} to zero velocity results in a collision (Bratt, Smith, & Christensen, 2006).

B. Requirement Analysis

In order to design a robot arm for our experimental setup, we start with an analysis of the requirements that need to be fulfilled. The experiment involves catching a ball thrown across a room. We anticipate a normal, slow, underhand throw from a distance of approximately 5 m. In an indoor environment, a ball can be thrown with reasonable accuracy along a parabolic path with an apex of 2.3 m, with both the thrower and the catcher situated at a height of approximately 1 m (see Figure 2). Simple studies of human performance indicates that the system must be able to accomodate variations in accuracy corresponding to catching the ball within a 60×60 cm window. From these basic requirements it is possible to compute the flight time and end velocity for the scenario, with the flight time being approximately 0.8 s, and the end velocity being approximately 6 m/s.

These experimental requirements in turn impose requirements on the platform. As stated in I, one of the desired

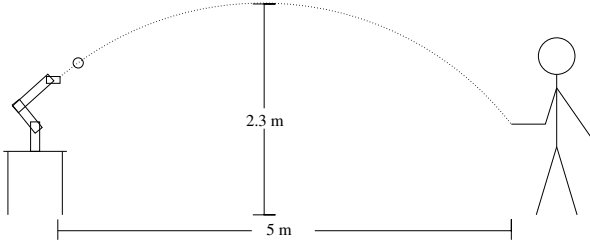


Fig. 2. Schematic of ball-catching experiment.

requirements of the scenario is use of standard video cameras for trajectory estimation. Using normal 50 Hz cameras, the frame time is approximately 20 ms, and a similar time window is expected to be needed for segmentation and position estimation. In addition, at least three frames are required for an additional trajectory estimation, resulting in a 60 ms estimation time. However, limited accuracy of the cameras will mean that more, probably as many as 10 images might be necessary (c.f. (Frese et al., 2001)), and the time delay from initiation of a throw to initial trajectory estimate might be 200 ms. This setup is intended to be used for teleoperated catching, so we also have to allow for extra reaction time as we include a human in the loop. This might add an additional 100 ms, so a window of 300 ms is reserved for initial reaction to a throw, leaving 500 ms in which the arm has to move into position. In the worst-case scenario, the arm has to move against gravity from one opposing corner of the operational window to another, a distance of almost 0.9 m. Depending on the type of end effector¹, the positioning has to be within one or a few centimeters of error from the ball trajectory. These requirements can be summarized as:

- End effector has to be able to move 0.9 m in 0.5 s, (partially) against gravity, from stand-still to stand-still.
- The precision of positioning the end effector should be within 1 cm

Given constant acceleration and deceleration, a distance of 0.9 m can be travelled in 0.5 s if the acceleration is at least 14.4 m/s^2 , and the maximum velocity is at least 3.6 m/s. This also has to be achieved when working against gravity. These are the minimum dynamic requirements — the actual implementation should have some margin to allow for uncertainties.

The system thus requires significant dynamics and the control has to be performed in real-time. This implies that it is desirable to have closed form solutions for kinematics, which in term imposes constraints on the design of the overall kinematic structure. Without a closed form kinematic/dynamic solution it would be much more challenging to guarantee the real-time performance.

A highly dynamic mechanical arm will pose a potential hazard to both its operator and itself unless sufficient precautions are taken. Therefore, the control of the arm has to be sufficiently exact so that safe paths can be accurately followed, and precautions against malfunctions have to be duly taken. The former requires control loops running at a high

frequency/low latency, the latter that software and hardware malfunctions are kept at a minimum, and that the negative effects of malfunctions also should be minimized. Thus, the software environment has to be a stable real-time system, while the hardware contains fail-safe fallback for dealing with software failure. These further requirements a solution has to fulfill can be summarized as:

- Closed form analytical kinematics and dynamics.
- At least 6 degrees of freedom.
- Acceleration of at least 14.4 m/s^2 for end effector.
- Velocity of end effector of at least 3.6 m/s.
- Safety for operator and machinery requires a stable real-time system, as well as fault-tolerant hardware.

C. Mechanical Design

There are a number of fairly fast robotic manipulators available, like for instance the DLR developed Kuka Light Weight Arm (Hirzinger, Sporer, Albu-Schafer, Haahnle, & Pascucci, 2002). It has been shown to be fast enough to catch thrown balls autonomously (Frese et al., 2001), but in order for the arm to reach the catch position in time, the ballistic path estimation has to be done very early. In our experiments we also want to include a human operator in the control loop to be able to do semi-autonomous teleoperated catching, so we require even faster movements to compensate for slow human reactions. With perhaps only half the time to get into position, twice the speed is needed.

Given the special needs of our experiments, it was decided to construct a 6 DoF arm using *PowerCube* modules from the German manufacturer Amtec. These modules are available off the shelf and allow for rapid prototyping. In addition, the modules have a built-in controller that can be used for embedded safety and low level control (C. Smith & Christensen, 2007).

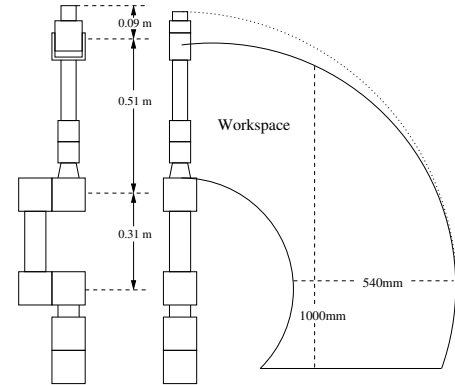


Fig. 3. The manipulator, constructed with Amtec PowerCubes.

The modules were assembled in a configuration that kinematically is very similar to a Puma560 arm (and to many other industrially available ones). This is not only a configuration that allows for very good dynamic performance but as it is a widely used and studied configuration, several implementation issues have already been solved, thus making the design process considerably faster. For example, the closed form solution for inverse kinematics and dynamics are well-known.

¹Since the initial experiments will not be concerned with grasping, a simple passive end effector like a net will be employed.

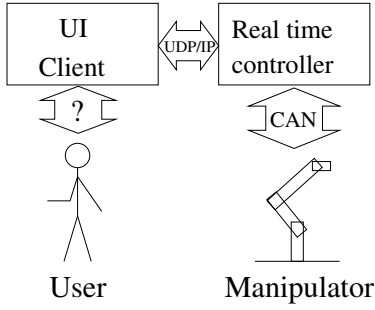


Fig. 4. Schematic of the connection architecture.

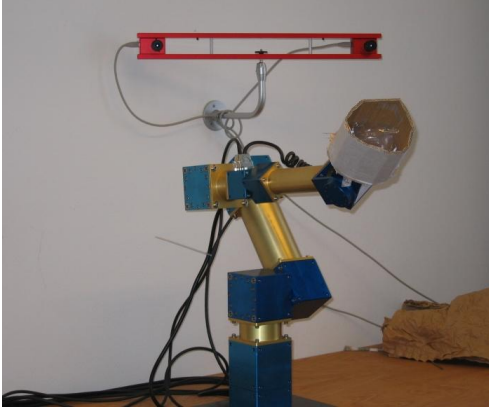


Fig. 5. The robot arm.

Keeping the moments of inertia as low as possible in the moving parts, and placing heavier, more powerful modules where their impact on the inertial load is lower, very fast dynamics can be achieved.

The design and dimensions of the configuration can be seen in Figure 3. The design allows for a workspace that is more than large enough to accommodate for the specified 60 cm \times 60 cm window for ball-catching, though the manipulator's dynamic performance deteriorates somewhat at the edges of the workspace. A cross-section of the workspace can be seen in Fig. IV-C. The arm has rotational symmetry as viewed from above.

It was decided that the computer doing the direct control should run RTAI, a real-time Linux system that has showed good performance in previous studies (Aarno, 2004). The control computer will perform the trajectory generation and be responsible for dynamic and kinematic calculations. A secondary computer will be used for the user interface. The communication between the two should be in cartesian space, since the kinematic structure of the arm allows for up to eight different joint-space configurations for each cartesian position, and the choice of configuration should be made by the real-time low-level controller for best performance. In early experiments over the LAN in our lab, the total roundtrip time from the UI input via the manipulator controller to UI feedback has been shown to be in the range of 10–20 ms. A schematic of the connection architecture is shown in Figure 4.

D. Control System

A Linux 2.6.9 Kernel was patched with RTAI 3.2 for low-latency real-time performance. A customized communications API was also implemented to guarantee low-latency communication with the PowerCube modules, as well as customized libraries for fast vector manipulations optimized for calculating arm dynamics. The control loop is run in soft real-time. Experiments have shown that this gives a worst case latency of less than 1 ms, which is sufficient. The average jitter for the control algorithm is 60 μ s, which is significantly less than the modules' latency of up to 600 μ s.

Inverse kinematics and dynamics are calculated using a C implementation of the analytical solution for a Puma arm (Craig, 1986), and the forward dynamics are calculated using the second algorithm proposed in (Walker & Orin, 1982). As a result, inverse kinematics can be calculated in 1.7 μ s, and dynamics in 41 μ s, so that all calculations needed in the control loop take less than 50 μ s. This means that virtually all latency in the control loop originates from the CAN bus communication path and the PowerCube modules response times.

Combined position and velocity control has been implemented on the system using a combined feed-forward computed torque control (CTC) scheme and a feed-back proportional integrating (PI) controller. When a new setpoint enters the controller, a velocity ramp trajectory is calculated in joint space. This trajectory is limited by a preset top velocity and a maximum acceleration. The limits on velocity and acceleration are chosen to limit the mechanical stress on the system, while being able to reach any point in the workspace in less than 0.5 s.

E. Camera Placement

To perform the manipulation task described in the previous section, a vision system capable of tracking the ball with sufficient accuracy to make early predictions of the ball trajectory is necessary. Given manipulator speed and the size of the catching area of the end effector, the ball should be safely caught if the trajectory can be estimated with an accuracy, at the catch point, which is better than 4 cm after it has flown for less than 0.3 s.

The available vision system consists of a wall-mountable stereo camera pair with a 60 cm baseline, see Fig 5. The cameras connect via Firewire to a Dell Precision 9510 workstation with a Pentium D dual core processor at 2.8 GHz. This setup allows for color images taken at 50 Hz at a resolution of 320 \times 240 pixels. The cameras have a field of view of approximately 50° in the horizontal plane.

We want to determine the optimal camera placement given these parameters. If possible, we want to place the cameras close to the manipulator, to simulate an antropomorphic viewpoint, so it is of interest to find how this would affect performance. It is well known that near-optimal performance for tracking and predicting trajectories of ballistic objects can be achieved with an extended kalman filter (EKF) (Welch & Bishop, 2004). It is therefore reasonable to adopt this approach and we define our measure of success S as the fraction of

trajectories that can be predicted to within 4 cm accuracy with an EKF (see Eq 3).

$$S = \frac{n_4}{n_{tot}} \quad (3)$$

In the above equation, n_4 is the number of trajectories predicted to within 4 cm, and n_{tot} is the total number of trajectories considered. Given the restrictions of our setup, mainly the ceiling height of our lab and the field of view of the cameras, it is not possible to see more than a small portion of the trajectory with an above view, so we are limited to camera placements in the horizontal plane. Therefore, the placement of the cameras can be parametrized with one parameter, the angle between the optical center of the cameras and the horizontal direction of the trajectory, θ (see Fig 6).

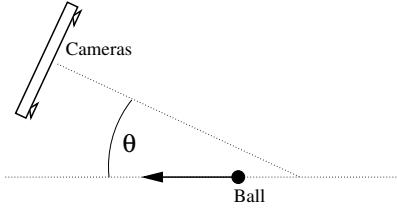


Fig. 6. Angle definition for camera placement

Thus, we are looking for a mapping that will give us S as a function of θ . Since no analytical solution is to be expected, the most straightforward approach to solving the problem is a Monte-Carlo type simulation of the ball-camera system. Simulating the system, it was possible to directly calculate S for a series of different values for θ .

Figure 7 shows the success rate measure S as a function of camera placement angle θ and elapsed flight time, for $\epsilon = 0.5$. Interception takes place after 0.95 s in these simulations. As can be seen, the performance is worst at 90° . Overall, the 180° position gives a good performance after a few images, but on the other hand, the final performance is good for the 0° position, and the success rate S has reached 0.75 after 0.4 s, meaning that 75 percent of throws would be caught using the prediction when 0.5 s remains.

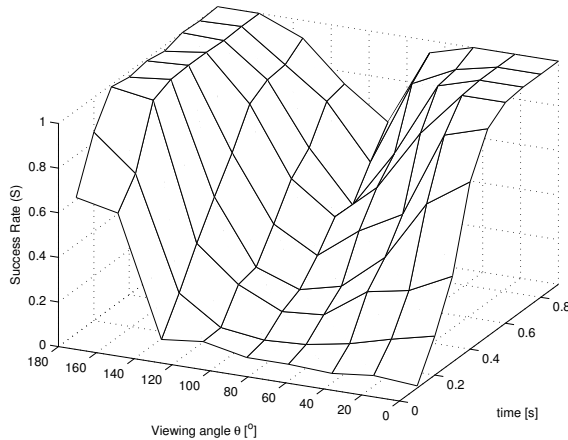


Fig. 7. Capture results with imaging error $\epsilon = 0.5$ pixels

Another way of viewing the results is to examine the variance of the Kalman state, P . If we eliminate measurements from the EKF iterations at time t_e and run the filter without new measurements to the intercept time t_i , the values of P at time t_i will give us the variance of the estimated trajectory at the point of intercept. We set t_e to 0.4 s in order to examine the results when 0.5 s remain. In table I, we give the standard deviation for the location of the ball at $t_i = 0.95$ s for some of the values of θ . The standard deviation is expressed along the axis of the trajectory, as well as the axis of the intercepting plane. Not only is there a difference in the magnitude of the error, but also in the orientation of the main axes. For $\theta = 90^\circ$, the error is smaller in the component in the direction of the trajectory, meaning that we can make a more exact prediction of *when* the ball intercepts the plane than *where* it intercepts it. For $\theta = 0^\circ$ or 180° , the situation is reversed. For our particular task, since the manipulator only has to be in position before the ball arrives, the exact timing is not important, and therefore we get better results with the cameras in the plane of motion, i.e., $\theta = 0^\circ$ or 180° .

TABLE I
STANDARD DEVIATION AT THE INTERCEPT TIME FOR $\epsilon = 0.5$

θ	along trajectory	intercept plane x	intercept plane y
0°	7.24 cm	1.16 cm	4.19 cm
90°	2.94 cm	19.39 cm	2.79 cm
180°	2.25 cm	0.6 cm	0.95 cm

The ball is detected in each image using a simple color segmentation scheme. First, the 24 bit RGB image was converted to 24 bit HSV using a lookup table. The ball was found to have a *hue* value of 3, and a (largely varying) *saturation* value of approximately 160, so all pixels that were in the range 1–5 for hue and 120–200 for saturation were preliminary marked as ball pixels. A second pass that only kept marked pixels with at least 3 other marked neighbors eliminated noise. The center of mass for the marked pixels was calculated and used as the ball centroid.

A variant of a subwindowing scheme as described in (Ishii & Ishikawa, 2001) was used to reduce the size of the processed image and thus the necessary processing time. After the ball has been detected the first time, only a subwindow where the ball should be expected to be found was processed. This subwindow was calculated using the state estimate from the EKF, and the size of the window is set to cover several times the standard deviation in position. Using this approach, the ball could be segmented and localized with a reasonable accuracy at less than 4 ms processing time per stereo image pair, giving sufficient real-time performance.

The evaluation was performed using 115 throws with the ball launcher. Since the accuracy of the vision system is within a few millimeters at the end of the trajectory, the last reading of the vision system was used as “ground truth” when evaluating the predicted trajectories. The resulting value for S as a function of time can be seen in Fig 8.

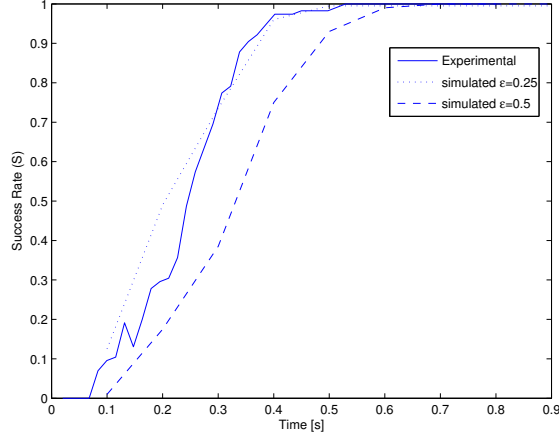


Fig. 8. Experimentally evaluated $S(t)$ for $\theta = 0$. The simulated results for the same angle, with $\epsilon = 0.25$ and 0.5 are also included for comparison.

F. Internet Communication Delays

The communication medium between the operator interface and the robot is the Internet or another IP network. Whereas this gives several advantages, including world-wide availability and abundant hardware and software support, it also poses some problems. There are no bandwidth guarantees, and, most importantly, there is a stochastic communication delay.

The magnitude and distribution of the delay (Niemeyer & Slotine, 2001) depends on several factors, such as physical distance, time of day etc. Typical round trip times are about 50 ms inside a city, 100 ms US coast to coast, and 150-200 ms for a transatlantic connection. Different kinds of fluctuations in the delay include random noise, changes due to variable network load, delays from transmitting too much data, bunching of packets not originally transmitted together, and lost packets (Niemeyer & Slotine, 2001). Improved real-time communication properties can be achieved by using IPv6 where available (Nuño & Basañez, 2005).

On top of the IP protocol, UDP (user datagram protocol) is used for the teleoperation communication. For transmission of real-time data UDP is preferable to TCP (transport control protocol), because it avoids the overhead of e.g. detecting and retransmitting lost packets, at the expense of not guaranteeing transmission of all data or constant ordering (Munir & Book, 2002). If the packets are self contained, i.e. they each contain samples of all the transmitted signals, only the newest packet available at the receiving side is relevant, making retransmission of old data and enforcing constant ordering pointless.

To avoid the problems caused by variable delays, the teleoperation system uses a constant virtual delay equal to a postulated maximum communication delay as detailed in section IV-A above. The virtual delay is realized by embedding a time-stamp in each packet and adding an artificial delay at the receiving side so that the sum of the communication delay and the artificial delay is equal to the virtual delay. It is chosen to make occasions when the communication delay is larger than the chosen value relatively rare. They can then be treated as exceptions by the control system causing the robot to stop



Fig. 9. One of the two user interface hardware configurations, with a CRT stereo display and an Omega haptic device.

smoothly.

G. User Interface

Two different versions of the hardware part of the user interface have been tested:

1. *Stereo display on a CRT monitor and a 3D force reflectance joystick.* Shutter glasses (from an inexpensive eDimensional 3D Vision System kit) are used for stereo, and the joystick is an Omega unit from Force Dimension. It is a parallel linkage device, which provides a large workspace, as well as high stiffness and force output. This user interface hardware is shown in Figure 9.
2. *Head mounted stereo display, headtracking and tracking of hand position.* In this configuration the display is an eMagin 3DVisor Z800 aimed at the high-end gaming market. It has one 600×800 OLED display per eye, and a diagonal field of view of 40 degrees. For headtracking and tracking of free hand motion an Ascension Technology Nest of Birds device is employed. To get acceptable quality headtracking we use a setup with two of the four available sensors fixed to the user's head and the magnetic transmitter unit mounted at a distance of only approximately 30 centimeters from these sensors. This allows measuring the orientation and position of the head with the in-device filtering disabled. It otherwise causes unacceptable lag and, consequently, nauseates the user. The two remaining sensors are bundled together and held by the user, so that hand position can be measured. The resolution of the hand measurements are reduced due to the greater distance from these sensors to the transmitter.

Both of these hardware setups use the same virtual environment, but only the second allows the user to change the gaze angle and viewpoint in a natural way. Both also use a speaker connected to the sound card of the user interface computer for audio feedback.

The basic visualization is performed using Open Inventor 6 from Mercury Computer Systems, and the force reflectance is

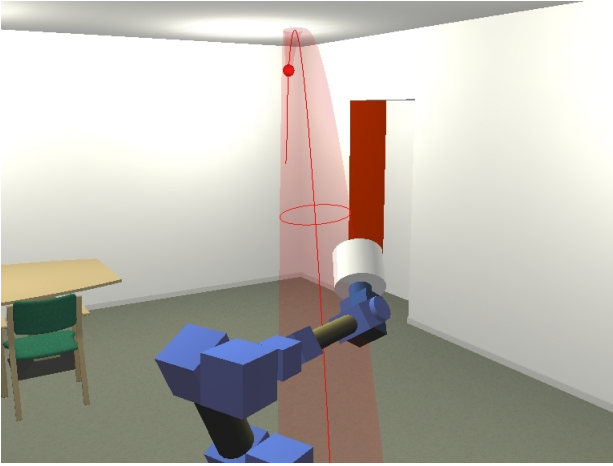


Fig. 10. The user's view in the ball-catching experiment.

implemented with the low level API provided by Force Dimension. Even though the Omega device setup and the software developed supports feeding back robot arm momentum and obstacle avoidance forces to the user, this functionality was disabled for the catching experiments described in Section V.

OpenGL quad buffered stereo with asymmetric frustum perspective mapping is supported, and the dependence of teleoperation performance on any static or dynamic virtual camera position can be explored. A screen dump of the operator interface is shown in Figure 10.

The user interface receives the estimated position of the ball from the stereo vision system of Section IV-E. In order to be able to bridge communication time delays using prediction as detailed in Section IV-A however, it also needs the complete state of the Kalman filter to make future ball position estimates. Estimates are computed by letting the state develop further forward in time without the usual measurement update. This predictive power is also used over the much longer time interval of a complete throw to show the user the projected trajectory of the ball.

The predicted ball trajectory is drawn as a red line through the ball, and the uncertainty inherent in the future Kalman states is displayed as a semi-transparent shell around it. The shell cross section that corresponds to a particular point on the trajectory is the projection along the predicted velocity of an uncertainty ellipsoid. The axes of the ellipsoid are twice the standard deviations along different directions derived from the eigenvectors and eigenvalues of the Kalman filter covariance matrix for the predicted x , y , and z coordinates of the ball. The result is that the uncertainty of the ball trajectory drawn is visible as a funnel that has its narrow end at the current ball position, and widens as the standard deviation, represented by the distance from the trajectory in the center, increases the further into the future the prediction extends.

When the ball is not in flight, but is visible to the vision system, the user interface computer still gets information about the estimated ball position. In this case, which often occurs just before a throw, the rest of the Kalman state is not available, and the ball is shown in an alternate color. As the vision system determines that a throw has begun, full Kalman state data starts

to flow in, and the word 'go' is output to the speaker to alert the user. Another sound is played if and when the ball is caught.

All data about user actions, as well as received robot and ball states, are logged to file by the user interface computer. The log files can then be exported and used for data analysis with tools such as Matlab, but also serves as input to the interface software in replay mode. This mode of operation allows normal and reduced speed play-back and pausing of recorded experiment sessions. The viewpoint and viewing direction can be arbitrarily adjusted in real-time using a 3Dconnexion SpaceTraveler six degree-of-freedom input device.

V. EXPERIMENTAL DESIGN

To evaluate the system, a pilot study comprising of a series of experiments was performed. The purpose was to examine how a user interacts with the system, to find what validity the human motion models have for predicting user inputs, and to determine if the setup is appropriate for tele-operated catching. In total, three experiments were conducted. The results from the first experiment were used to guide the design of the latter.

A. Experimental Procedure

The main goal of the first experiment was to study the interaction between the user and the system, and to evaluate the user interfaces. The goal of the second experiment was to evaluate the online user input prediction ability of the MJ model, and the ball-catching performance of the system. The goal of the third experiment was to verify the effect of user instruction.

1) *Experiment 1 - User Interfaces:* In the experimental setup, the same virtual world was presented in the two different interfaces presented in Section IV-G. The user's viewpoint was slightly behind and above the manipulator with a clear view of the rest of the room and the path of the incoming balls, see Figure 10. Balls were visualized as small spheres, and the balls' ballistic trajectories were recordings from earlier tries with the autonomous ball-catching system.

In these trials, only the simulated virtual reality system was used. Since the goal was to study the interaction between the user and the interface, it was deemed that connecting the actual robot arm would not benefit the experiments.

The subjects were given a brief explanation of how the interface works. They were told that they would see balls being thrown towards them, and that they were to try to steer the robot arm so as to intercept the ballpath and thereby catch the ball. With setup 1, the subjects were instructed that the arm would copy the movements of the input device, and were given some time to explore this before the subject of ball-catching was brought up. In setup 2, subjects were told that the robot arm would mimic the motions of their right hand (or left for the one left-handed subject), and that the end effector of the manipulator would be visualized in the same place in the virtual world as their hand would be in the real world.

For each system, the subjects were given 20 practice throws, in which the balls were thrown in a random order. After a short break, they were subject to 21 throws that they were told would be measured. When using setup 2, the subjects were allowed

a short rest after each 7 throws, dividing the experiment into three shorter sessions. This was so that they could rest their arms/shoulders.

A total of 10 subjects were used, all of whom are robotics or computer vision researchers affiliated with our university, but with no prior knowledge of the setup. Of these 3 were female and 7 were male. The same subjects were used for both setups, but the trials with the different systems were conducted approximately two weeks apart. The trials with setup 1 were performed before the trials with setup 2.

2) *Experiment 2 - Online Prediction:* In the second experiment, interface setup 1 was connected to the robot manipulator over a UDP/IP connection, using the control structure described in Section IV-F. A mechanical ball launcher was set to launch balls in a predetermined pattern, that was the same for all subjects, but unknown to the subjects. In this experiment, the communication roundtrip delay was about 20 ms, caused mostly by the 50 Hz operational frequency of the communication protocol.

The goal of this experiment was to study how online MJ prediction can be used to aid the user. The final objective is to design a system that can use prediction of user inputs to bridge delays in the communication link. Three different systems for sending user input to the manipulator were employed:

- **System A - Direct Control** In the direct control setup, position and velocity data from user input was transmitted directly as setpoints to the robot controller, at 50 Hz. The controller would then accurately mimic the user's movements up to the maximum acceleration of the robot.
- **System B - Minimum Jerk Prediction Based Control** The minimum jerk (MJ) prediction based control system fits MJ trajectories to the user input data whenever a velocity threshold is exceeded. This was empirically set to 0.07 m/s. The MJ model is then used to predict the goal in time and space of the current user motion. A prediction is made when a velocity peak has been detected, leaving the second half of the motion to be predicted using data from the first half. The predicted goal of the user motion is sent as a setpoint for the robot arm instead of the current motion.
- **System C - Semi-Autonomous Control** The semi-autonomous system is based on system B, and works identically, until a predicted user input is found to be within 10 cm of the expected ball trajectory, when the user is assumed to attempt to catch the ball, and an autonomous ball-catching system is engaged. The autonomous system sets the robot controller setpoint to the point on the expected ball trajectory that is the closest to the present position.

Each subject was first shown the simulator version of the teleoperation system to allow him or her to get acquainted with the user interface. The relevant features of the graphics display were explained. These include a home position that the end-effector must be brought to before each throw, the ball flying toward the robot, and the projected trajectory of the ball. Then the subject was allowed 20 practice throws with simulated robot dynamics and prerecorded balls.

The following sequence was repeated for each control system:

- Twelve unrecorded practice throws followed by a pause while the balls were collected.
- Three times twelve recorded throws with a pause after each series of twelve.

The order in which the systems were presented was permuted between subjects as to not introduce a bias from learning curve effects. The subjects were not informed of the underlying control model for each of the systems. A total of 25 subjects were used. These had no prior experience of robot control. 12 were female, 13 male, and the ages ranged from 19 to 63 years. All the subjects were persons unaffiliated with the university that had responded to an advertisement.

3) *Experiment 3 - User Instruction:* The purpose of the third experiment was to examine what effect user instruction would have on performance. The experiment was in most parts identical to experiment 2, with three differences. First, only systems A and C were employed — system A to provide a performance baseline, and system C to see if assistance would benefit a user when properly instructed. System B was omitted as it was not expected to enhance performance in the absence of time-delays.

The second difference was that the users were briefly instructed how the systems worked, in terms resembling the system description in the previous section. The third difference was that a visual cue was added so that the end effector of the manipulator changed color in the graphical interface when the autonomous system was engaged. For this experiment 10 subjects aged 25 to 37 were used. Of these, 2 were female and 8 were male. These subjects had no prior experience with the experimental setup.

B. Trajectory Estimation

From the collected data, we attempt to predict the subjects' reaching trajectories using data from the start of the trajectory. Using current findings on human catching motions, it should be possible to some extent to predict the final position of a motion only using a portion of the start of that motion.

In our implementation, we assume that our subjects reach towards an expected interception point along a minimum jerk (MJ) trajectory, and that corrections of the initial motion are carried out by superimposing additional MJ trajectories onto the first, a model described by Flash et. al (Flash & Henis, 1991; Flash & Hogan, 1985).

If we know the duration of a motion, Δt , we can use this to fit the motion to a MJ trajectory, which according to the model accounts to a 5th degree polynomial with boundary constraints, Equation 2.

We assign t_0 to the start of the motion and set $t_1 = t_0 + \Delta t$. If we now assume that the motion stops at t_1 , meaning that both velocity and acceleration are zero, we get the constraints in Equation 4.

$$\begin{aligned}\dot{x}(t_1) &= 0 \\ \ddot{x}(t_1) &= 0\end{aligned}\tag{4}$$

Combining equations 2 and 4, we get Equation 5.

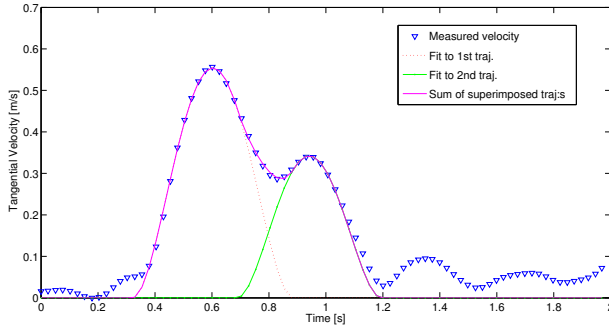


Fig. 11. A fitting of two superimposed 4th-degree polynomials to a measured tangential velocity profile

$$\begin{aligned} \mathbf{x}(t) = & (t^5 - 10t^2t_1^3 + 15tt_1^4)a_1 + \\ & (t^4 - 6t^2t_1^2 + 8tt_1^3)a_2 + \\ & (t^3 - 3t^2t_1 + 3tt_1^2)a_3 + \\ & a_6 \end{aligned} \quad (5)$$

1) *Offline Trajectory Fitting*: For experiment 1, user data was collected and analyzed offline. As the subjects were given several tries to practice, we also assume that the initial reaching motion will be of approximately equal length in different trials with the same person, as the velocity of the approaching ball and the distance to the intercept point is similar between trials. Our first approach was therefore to extract the average time for a reaching motion.

In order to find the time Δt , it is important to identify where one MJ trajectory starts and ends. Since the total motion can contain several MJ trajectories superimposed on each other, they must be separated. Our method for separation works by identifying the typical bell-shaped tangential velocity profiles of a motion (Milner, 1992; Flash & Hogan, 1985). These bell-shaped velocity profiles are 4th-degree polynomials with boundary value conditions that state that they start and end at zero, with zero derivatives. This leaves only one free parameter, which can be fit using a least-squares approach. Since we anticipate several superimposed trajectories, we fit several such 4th-degree curves, minimizing the mean square error of the sum of these trajectories as they approximate the total velocity profile. An example of the results of such a fitting is shown in Figure 11. We assume that submovements are at least 100 ms apart, as observed by Milner (Milner, 1992).

We extract the average duration of the first motion of each recording, and use for Δt . Using the measurements from experiment 1 in Section V-A1, we fit Equation 5 to the data using least squares fitting. The parameter t_0 was determined by finding the point in time after a throw commenced that the motion exceeded a predefined threshold velocity v_0 .

The less data that is used for fitting the MJ trajectory, the earlier a prediction can be made. However, it is expected that the quality of the prediction will deteriorate with the use of fewer samples. It is therefore of interest to find what performance can be expected when using different portions, α , of the observed trajectory data before making a prediction.

TABLE II
CATCHING SUCCESS RATES

Subject	Setup 1	Setup 2
1	0.62	0.71
2	0.24	0.33
3	0.29	0.38
4	0.48	0.67
5	0.48	0.52
6	0.29	0.10
7	0.76	0.71
8	0.81	0.62
9	0.71	0.33
10	0.33	0.43
total	0.50	0.48

2) *Online Trajectory Fitting*: For the subsequent experiments, online fitting was necessary. Observing that the typical bell-shaped velocity profile of a MJ motion is symmetrical in time around the point of maximum velocity, we see that we can use the fourth degree fitting scheme mentioned above using only the first half of the motion. Thus, it is possible to find the start point and predict the endpoint of a movement when half of that movement has been performed. Therefore, for the online curve fitting, α was set to 0.5. The midpoint of a motion is defined as the point where the second derivative, acceleration, changes sign from positive to negative. By using an extended Kalman filter, this point is easy to detect. Using a least square approach, the 4th degree velocity profile can be fit around this peak, fitting only to data before the peak. The zeros of the polynomial found with this approach are used as candidates for t_0 and t_1 for the motion, and used when fitting the motion's position data to a MJ trajectory.

VI. EXPERIMENTAL RESULTS

This section presents the results of the experiments. When making comparisons between the simulation in experiment 1 and the performance of the real robot setup in experiments 2 and 3, it should be noted that balls are much easier to catch in the simulated environment, mostly due to the lack of modelling of collision dynamics. Thus, all balls that hit the catching cylinder are considered to be “caught” in the simulator, while many of these might bounce off in the real system.

A. Experiment 1 - User Interfaces

1) *Catching Performance*: In the simulation experiments, the catch performance varied largely between subjects. Some subjects were able to successfully catch most balls after the initial practice, while others were hardly able to catch any. The success rate for the different setups are shown in Table II. According to an early evaluation, there seems to be a correlation between successful stereo fusion in the interface and successful catching.

2) *Minimum Jerk Fitting*: To illustrate the fitting of a MJ trajectory to recorded data, two examples are shown. Both examples originate from subject 1, for whom the performance of MJ fitting was average on setup 1.

The first example, in the left part of Figure 12, illustrates a catching motion that was close to the MJ model, and could

be fit well to a MJ trajectory. The plot shows the tangential velocity profile at the top, and the different components below. The green dotted line is the fit MJ trajectory, ended with an asterisk. The first vertical line represents the start of the motion, and the second vertical line shows the last sample used for fitting. In this example, a portion of $\alpha = 0.35$ of the motion was used for prediction. The second example, in the right part of Figure 12, shows the performance when a large corrective motion is added after the initial motion. As can be seen, this motion is not a good fit to the MJ trajectory, mostly due to the fact that almost no samples of the corrective motion are used for fitting the MJ trajectory.

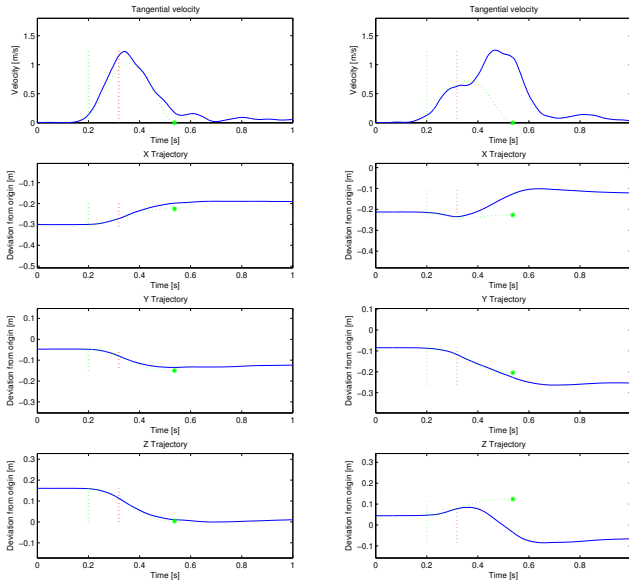


Fig. 12. Examples of fitting a MJ trajectory to a catching motion using the first 35% of the motion to fit. The left plots show a successful fit, and the right plots show a failed fit. The solid line is the measured trajectory, the dotted line is the MJ trajectory fit. The two vertical dashed bars enclose the portion of the trajectory used for fitting. The portion of the MJ trajectory that continues beyond this portion is purely extrapolated.

3) *Results Using Setup 1:* The following results were recorded from the users in the trial with setup 1. The average time and average traveled distance for the the first MJ trajectory for the different subjects are presented in Table III. The distances are measured in the robot space for easy comparison. The actual distance moved with the device is 10 times smaller. Given that all subjects try to catch balls from an identical set of trajectories, it could be expected that the differences in traveled distance would be smaller. Some subjects would return the robot to a default position between catches. This position varies across subjects, and accounts for most of the difference in average distance.

The average deviation from final position for the fitted MJ trajectories is shown in Figure 13. This was calculated by using the samples from the first portion of the motion to fit an MJ trajectory, and then measure the distance between the end of the predicted trajectory, and the actual position at this time. When only a small portion ($\alpha < 0.2$) of the samples

TABLE III
AVERAGE TIMES AND DISTANCES FOR MJ MOTIONS ON SETUP 1

Subject	Avg Δt [s]	Std(Δt) [s]	Avg distance [m]	Std [m]
1	0.3376	0.0859	0.2115	0.0859
2	0.2974	0.0462	0.5753	0.1078
3	0.3906	0.0498	0.3638	0.1216
4	0.3076	0.0940	0.1806	0.0692
5	0.3883	0.0853	0.2594	0.0678
6	0.3059	0.0948	0.3552	0.1283
7	0.3190	0.0839	0.1945	0.0724
8	0.2874	0.0863	0.1935	0.0565
9	0.3738	0.0876	0.4557	0.0731
10	0.3471	0.1015	0.2348	0.1382
total	0.3355	0.0801	0.3024	0.1571

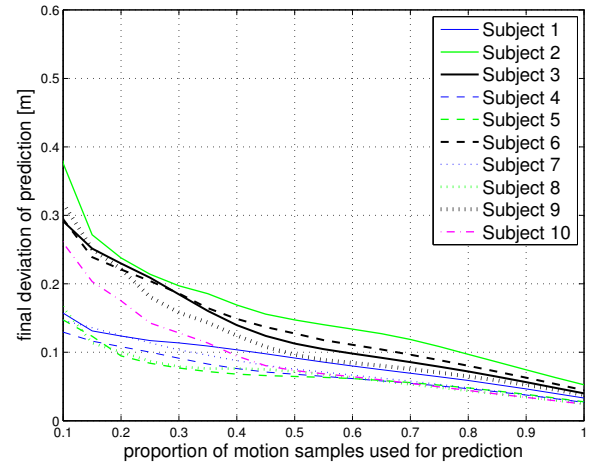


Fig. 13. The final deviation of the predicted MJ trajectory as a function of the portion of the trajectory used for fitting, using setup 1.

were used, insufficient data accounts for the larger part of this deviation. When more samples are used, non-modeled corrective movements after the used samples account for most of the deviation.

To show the performance of MJ trajectory fitting, the results are compared to other, simpler models. For MJ fitting to be meaningful, it should at least outperform simpler approaches that do not use human motion models. Perhaps the simplest fitting possible is to use the last available sample and perform a constant extrapolation from this. Since this model does not assume any motion after the last observation, it can be viewed as a zero-order model. If motion is assumed, using least square fitting to fit linear, quadratic, or cubic functions to the data set can be used. A comparison of the performance of MJ trajectories and extrapolation of polynomials of different degrees is shown in Figure 14. Polynomials of higher degree than 3 result in even larger deviations when used for extrapolation. Note that in all these fittings, the MJ criterion has been used to determine the duration of the motion, which is essential to predicting the endpoint for all but the zero-order model.

4) *Results Using Setup 2:* Most of the discussion in the previous section is valid for these results as well. The average time and the average motion distances (in robot space) for the first MJ trajectory is presented in Table IV.

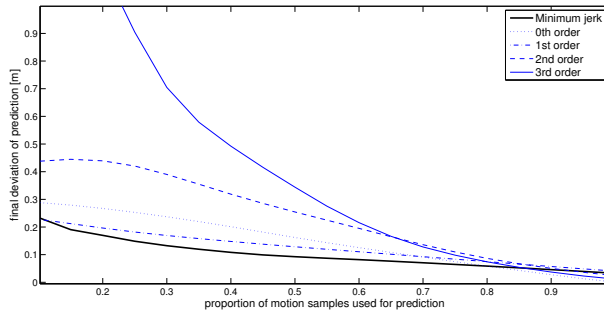


Fig. 14. The final deviation of the predicted MJ trajectory as a function of the portion α of the trajectory used for fitting, as compared to the final deviation when fitting polynomials, using setup 1. This is the average over all subjects.

TABLE IV
AVERAGE TIMES FOR MJ MOTIONS ON SETUP 2.

Subject	Avg Δt [s]	Std(Δt) [s]	Avg distance [m]	Std [m]
1	0.7229	0.1063	0.2579	0.1174
2	0.5268	0.0628	0.5082	0.1018
3	0.6738	0.1430	0.3190	0.1832
4	0.6799	0.1291	0.3987	0.2876
5	0.8269	0.1361	0.3135	0.0896
6	0.4860	0.0965	0.4211	0.1330
7	0.6703	0.1162	0.3315	0.1264
8	0.7474	0.1201	0.3971	0.0977
9	0.5945	0.1010	0.3774	0.1282
10	0.6112	0.1218	0.5137	0.1574
total	0.6542	0.1492	0.3838	0.1688

The average deviation from final position for the fitted MJ trajectories is shown in Figure 15. It should be noted that this setup has a significantly lower sampling frequency than setup 1, and therefore the results for using only the first 10 or 20% of the samples are not very reliable. A comparison of the performance of MJ trajectories and extrapolation of polynomials of different degrees is shown in Figure 16.

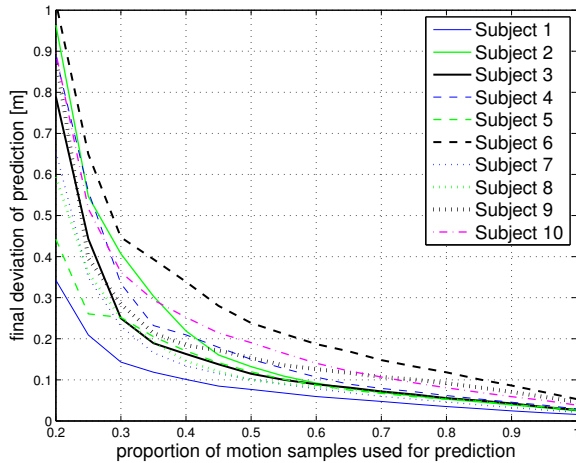


Fig. 15. The final deviation of the predicted MJ trajectory as a function of the portion α of the trajectory used for fitting, using setup 2.

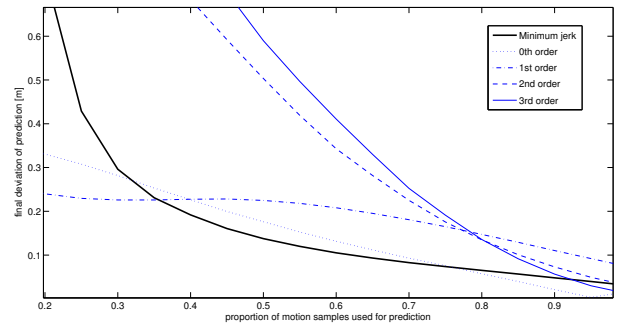


Fig. 16. The final deviation of the predicted MJ trajectory as a function of the portion α of the trajectory used for fitting, as compared to the final deviation when fitting polynomials, using setup 2. This is the average over all subjects.

5) *Result Analysis:* As can be seen from the plots of fitting performance, the MJ fitting outperforms the other fitting methods, at least for a portion of the interval. In both experimental setups it seems that MJ fitting is better when one to two thirds of the data is used for fitting. However, it is also obvious that this performance varies greatly with the subject. Some subjects seemed to have problems with immersion into the 3D environment. The subjects that were actually able to catch a significant portion of the balls reported good immersion with the interface whereas those with fewer caught balls indicated a weaker immersion.

There seems to be a connection between successful catching and successful fitting to a MJ trajectory. An explanation for this can be that subjects that perform well with the catching task are able to make a good early prediction of the ball trajectory and therefore need less corrective motions. This proposition is supported by the observation that many of these subjects tend to move shorter distances with less variation, as can be seen in Tables III and IV.

B. Experiment 2 - Online Prediction

The first performance measure considered is the performance of the online MJ predictor. The overall performance is summarized in Table V. The first MJ trajectory that has been found is evaluated. The average duration is given, and also the average improvement. The improvement factor is how much closer the predicted position at t_1 is to the later measured position at t_1 , compared to the remaining distance when the prediction is made. Thus an improvement factor of 0 would mean no improvement, and 100% would mean that the system had made a prediction that was exactly the same as the actual value. Note that predictions in the wrong direction give negative improvement factors. The number of tries for which the predictor improves the position by at least 25, 50, and 75% are also given.

Comparing the catching performance of system A, where the MJ predictions were not used to control the robot, but merely logged, with the MJ predictor performance, we find that good MJ prediction correlates to successful catching, with $p = 0.010$. If we only examine the successful catches from system A, we find that the average improvement of MJ

TABLE V
THE PERFORMANCE OF MJ PREDICTION IN EXPERIMENT 2.

	average	std
Movement Duration	0.249 s	0.0895 s
Relative Improvement	17.2%	82.3%

improvement	trials within ratio
25%	44.9%
50%	26.0%
75%	5.9%

TABLE VI
THE PERCENTAGE OF BALLS CAUGHT IN EXPERIMENT 2.

Trial	Balls Caught
System A	20.6%
System B	16.4%
System C	15.7%

predictions is 32.2%. There was also a significant correlation between the reaction time and the performance of the MJ predictor, with $p = 0.0124$. The faster the reaction, the better the MJ prediction.

The second performance measure considered is the number of balls that were caught with the different systems, as shown in Table VI. The figures here are the average over all tries in each experiment, i.e. 900 tries for each system in experiment 2. The unexpected result in experiment 2 is that subjects performed worse with the assisted system C than with the other systems.

For reference, the robot system was also run in fully autonomous mode without any user connection. In this case, the reaction time is on the order of 100 ms, and the initial error that needs to be corrected is minimal. The catching performance for this case was that 73% of the balls were caught, 19% bounced off the rim of the end effector and the remaining 8% were missed completely. This is the hardware limit on performance for the set of ball trajectories used in the experiments.

Defining the performance as the ratio of balls actually caught is an aggregate measure of the performance of human and machine. In order to discriminate between the subject's and the system's performance, each subject's logged input signals were examined. There are two major components of the quality of user input, the spatial precision of the input, and the reaction time, good input being both fast and accurate.

Since the spatial accuracy depends on the initial distance from the manipulator and thus the user's hand position to the ball trajectory (Fitts, 1954), we formulate the accuracy A as the ratio of the distance d_c when the ball was the closest to the manipulator and the initial distance d_i to the trajectory, as formulated in Equation 6.

$$A = \frac{d_c}{d_i} \quad (6)$$

With the system C, there was a tendency that the earlier a catch attempt was recognized and the autonomous system was engaged, the higher the probability for catching. This relationship is illustrated in Figure 17. The plot shows an

almost linear relationship between the distance left for the ball to travel and the probability for the ball to be caught.

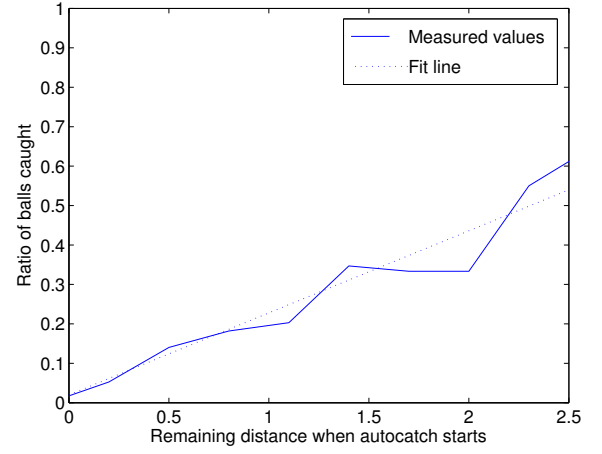


Fig. 17. The probability that a ball was caught as a function of the distance remaining to the interception point when the autonomous system was engaged. The dashed line shows a linear fit to the data.

One way to measure reaction time that is relevant to the current setup is to see when the intention detector in system C was able to detect the intention to catch the ball. Figure 18 shows the distribution of remaining distances to the interception point when the catch intention was detected.

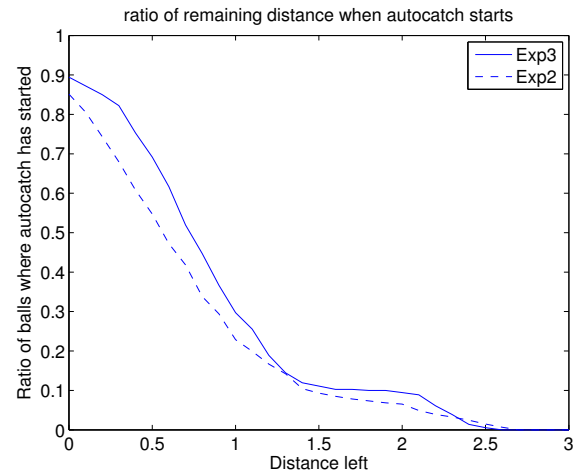


Fig. 18. The distribution of the ratio of trials where a catch attempt has been detected at different distances to the interception point.

In experiment 2, it was found that the subjects' inputs had noticeably lower accuracy for system C than for the other two systems. Therefore, the poor performance of system C was thought to be due to this poor performance in the user input. The distributions of accuracies for the different systems in experiments 2 and 3 are shown in Figure 19. Several subjects in experiment 2 complained that system C was difficult to understand and did not perform as they expected.

C. Experiment 3 - User Instruction

For experiment 3, the performance of the online MJ predictor is summarized in Table VII. The first MJ trajectory that has

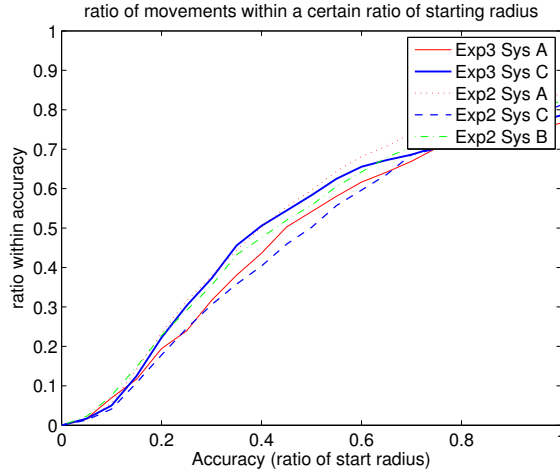


Fig. 19. The distribution of accuracies for the different trials. The plot shows the accumulated ratio of attempts that were better than a certain accuracy.

TABLE VII
THE PERFORMANCE OF MJ PREDICTION IN EXPERIMENT 3.

	average	std
Movement Duration	0.248 s	0.0875 s
Relative Improvement	19.6%	71.3%

improvement	trials within ratio
25%	46.0%
50%	23.9%
75%	5.8%

been found is evaluated. The average duration is given, and also the average improvement, as explained in the previous section. These figures do not differ much from experiment 2, except the average relative improvement which is slightly better, with a slightly smaller standard deviation.

As in experiment 2, there was a significant correlation between the reaction time and the performance of the MJ predictor, with $p = 0.0029$, but good predictions and successful catching in system A only correlates with $p = 0.14$, with an average improvement factor of 30.0% for successful catches.

The second performance measure considered is the number of balls that were caught with the different systems, as shown in Table VIII. The figures here are the average over all tries in each experiment, i.e. 360 tries for each system in experiment 3. The noteworthy result is seen in comparison with experiment 2: the number of caught balls varies with the information given to the subject. For system C, there is an increase in the number of balls caught when the subject is told that (s)he will be assisted by the system, while for system A, there is a smaller decrease in performance when the subjects are told that they will not be assisted.

Also, as can be seen in Figures 19 and 18, the user input

TABLE VIII
THE PERCENTAGE OF BALLS CAUGHT IN EXPERIMENT 3.

Trial	Balls Caught
System A	18.0%
System C	24.2%

performance is better for system C in experiment 3. Regarding reaction speed, since the velocity is close to constant, this figure can also be expressed in terms of time. For experiment 2, the median time left when a catch attempt was detected was 185 ms, in experiment 3, the median remaining time was 225 ms, which is 22% longer.

VII. EVALUATION

Though much remains to be studied, some early indications and observations are presented here.

A. User Interfaces

The two different user interfaces that were tried in the first experiment showed similar performance in terms of the number of balls caught. Given the full freedom of motion and high level of immersion in setup 2, this was expected to fit the MJ requirements better. However, the motion times were faster, and the prediction performance better for setup 1. Setup 1 is also less cumbersome to employ, and was therefore chosen for the subsequent experiments.

B. Online Prediction

On average, using the MJ model improved the knowledge of the final position by 17 to 20% in the two experiments. However, the variation between trials was large. Predictions to within 25% of the remaining distance was only possible in less than 6% of the trials. There were significant correlations between successful catching and fast reaction times and MJ prediction performance, which suggests that when a subject has a clear goal for the hand motion and can react in good time for a successful catch, the hand motion is a more typical MJ motion with small corrections. Manually inspecting examples of good performance, it seems that for most of these cases there is only one major MJ component, while for many unsuccessful attempts, several corrective motions as large as or larger than the first followed. Although this remains to be experimentally studied, it is possible that by using longer training sessions and thereby more skilled subjects, the number and magnitude of corrective motions would decrease, resulting in better performance for the MJ predictions.

The performance improvement resulting from informing the subjects that there is an assistive function in the teleoperation system indicates that using informed subjects is also an important feature to include in future implementations.

C. Future Directions

Summarizing the above, it seems relevant to have more user training before the MJ predictor model can perform well enough to serve its purpose of bridging communication delays.

VIII. SUMMARY

Traditionally people have studied teleoperation systems with limited dynamics. For such systems it is possible to have model based control system that compensate for delays and stochastic variations in the communication link. For systems with significant dynamics an interesting challenge is to decide

how to provide the user with effective ways of interacting with a system. I.e. how should the system be modelled in order to accomodate remote operation.

In human studies of ball-catching it is evident that a significant part of the motion is ballistic. Once a flying object is detected the arm is largely controlled ballistically to arrive at a position that is close enough for catching or for a small correction of the position. The present paper has studied teleoperation of catching as an example of how one might design system for operation with significant dynamics. This raises questions as to the fidelity of models of trajectory generation, appropriate mechanisms for feedback generation etc.

Human models for such systems were briefly outlined. In addition the design of a high-performance manipulator for experimentation was outlined and two different interfaces for control of such a system were presented. The system has been used in a number of early experiments for ball-catching. The experiments have been performed across 45 subjects and using both a joystick / 3D graphics interface and a virtual reality / body tracking interface. As a model for ballistic motion generation a minimum jerk trajectory has been suggested and also evaluated in this paper.

It is evident that model based control is the only way to achieve task performance in the presence of significant delays and/or dynamics. The task in itself seems to be difficult for untrained subjects, but when the subjects perform well, they also fit well with the suggested model, and there seems to be a good possibility for predicting the motions of skilled users. The system itself has a good enough performance to allow well above chance performance for the ball catching.

It is important to recognize that these results are preliminary results and in most cases they are not abundant enough to allow demonstration of significance. They should be taken as early demonstration of the value of human motion models. There are several open issues to be addressed in future research. This includes a need for clear modelling of the different types of users: novice, experienced computer gamer, tennis player etc. It is clear from the early results that there is a significant variability. In addition it is also clear that there is a learning effect which will influence the performance. Finally the time delays experienced this far are limited and it is of interest to study the performance as the delays increase and become more dominantly stochastic.

ACKNOWLEDGMENT

The research presented in this paper is funded by the 6th EU Framework Program, FP6-IST-001917, project name Neurobotics.

REFERENCES

- Aarno, D. (2004). *Autonomous path planning and real-time control - a solution to the narrow passage problem for path planners and evaluation of real-time linux derivatives for use in robotic control*. Unpublished master's thesis, Department of Numerical Analysis and Computer Science (NADA), KTH, Sweden. (TRITA-NA-E04006)
- Andersson, R. (1989, December). Dynamic sensing in a ping-pong playing robot. *IEEE Transactions on Robotics and Automation*, 5(6), 728–739.
- Åström, K. J., & Wittenmark, B. (1995). *Adaptive control* (2nd ed.). Boston, MA: Addison-Wesley.
- Berthoz, A. (2000). *The brain's sense of movement*. London, UK: Harvard University Press.
- Bratt, M., Smith, C., & Christensen, H. I. (2006, Oct). Design of a control strategy for teleoperation of a platform with significant dynamics. In *Iros-06* (pp. 1700–1705). Beijing, China.
- Craig, J. (1986). *Introduction to robotics: Mechanics and control*. Addison-Wesley Pub. Co., Reading.
- Dessing, J. C., Peper, C. E., Bullock, D., & Beek, P. J. (2005). How position, velocity, and temporal information combine in the prospective control of catching: Data and model. *Journal of Cognitive Neuroscience*, 17(4), 668–686.
- Fitts, P. (1954). The information capacity of the human motor system in controlling the amplitude of movements. *Journal of Experimental Psychology*, 47, 381–391.
- Flash, T., & Henis, E. (1991). Arm trajectory modification during reaching towards visual targets. *Journal of Cognitive Neuroscience*, 3, 220–230.
- Flash, T., & Hogan, N. (1985, July). The coordination of arm movements: An experimentally confirmed mathematical model. *The Journal of Neuroscience*, 5(7), 1688–1703.
- Frese, U., Bauml, B., Haidacher, S., Schreiber, G., Schaefer, I., Hahnle, M., et al. (2001). Off-the-shelf vision for a robotic ball catcher. In *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems* (pp. 1623–1629).
- Gat-Falik, T., & Flash, T. (1999, Feb). The superposition strategy for arm trajectory modification in robotic manipulators. *IEEE Transactions on Systems, Man, And Cybernetics-Part B: Cybernetics*, 29(1), 83–95.
- Goertz, R. C. (1952). Fundamentals of general-purpose remote manipulators. *Nucleonics*, 10(11), 36–45.
- Goertz, R. C., Blomgren, R. A., Grimson, J. H., W. M. Thompson, G. A. F. and., & Kline, W. H. (1961). The ANL model 3 master-slave electric manipulator - its design and use in a cave. *Transactions of the American Nuclear Society*, 4(2), 219–220.
- Harris, C. M., & Wolpert, D. M. (1998, Aug). Signal-dependent noise determines motor planning. *Nature*, 394, 780–784.
- Hauck, A., Sorg, M., & Schenk, T. (1999). What can be learned from human reach-to-grasp movements for the design of robotic hand-eye systems? In *Proceedings of the IEEE international conference on robotics and automation* (Vol. 4, pp. 2521–2526). Detroit, Michigan, USA.
- Hirzinger, G., Fischer, M., Brunner, B., Koeppel, R., Otter, M., Gerbenstein, M., et al. (1999, November). Advances in robotics: The DLR experience. *Intl. Jour. of Robotics Research*, 18(12), 1064–1087.

- Hirzinger, G., Landzettel, K., & Fagerer, C. (1994). Telerobotics with large time delays. In *Proc. of the intl. conf. on intelligent robots and systems* (pp. 571–578).
- Hirzinger, G., Sporer, N., Albu-Schaffer, A., Haahnle, M., & Pascucci, A. (2002). DLR's torque-controlled light weight robot iii - are we reaching the technological limits now? In *Proc. of the intl. conf. on robotics and automation* (pp. 1710–1716).
- Hogan, N. (1985). An organizing principle for a class of voluntary movements. *Journal of Neuroscience*, 4, 2745–2754.
- Hove, B., & Slotine, J. (1991, Jun). Experiments in robotic catching. In *Proceedings of the 1991 american control conference* (Vol. 1, pp. 380–385). Boston, MA.
- Ishii, I., & Ishikawa, M. (2001). Self windowing for high-speed vision. *Systems and Computers in Japan*, 32(10), 51–58.
- Kawato, M. (1996). Trajectory formation in arm movements: Minimization principles and procedures. In H. N. Zelaznik (Ed.), *Advances in motor learning and control* (pp. 225–259).
- Kosuge, K., Murayama, H., & Takeo, K. (1996, Nov). Bilateral feedback control of telemanipulators via computer network. In *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems* (pp. 1380–1385). Osaka, Japan.
- Krüger, N. (2006). *Entwicklung einer humanoiden ball-fang-strategie für ein roboter-hand-arm-system*. Unpublished master's thesis, Technischen Universität München.
- Kyriakopoulos, K. J., & Saridis, G. N. (1988). Minimum jerk path generation. In *Proceedings of the IEEE international conference on robotics and automation* (Vol. 1, pp. 364–369). Philadelphia, PA, USA.
- Land, M. F., & McLeod, P. (2000, Dec). From eye movements to actions: How batsmen hit the ball. *Nature Neuroscience*, 3(12), 1340–1345.
- Marken, R. S. (2005). Optical trajectories and the informational basis of fly ball catching. *Journal of Experimental Psychology*, 31(3), 330–634.
- McBeath, M. K., Shaffer, D. M., & Kaiser, M. K. (1995, Apr). How baseball outfielders determine where to run to catch fly balls. *Science*, 268, 569–573.
- McIntyre, J., Berthoz, A., Zago, M., & Lacquaniti, F. (2001). Does the brain model newton's laws? *Nature Neuroscience*, 4(7), 693–694.
- McLeod, P., Reed, N., & Dienes, Z. (2002, Dec). The optic trajectory is not a lot of use if you want to catch the ball. *Journal of Experimental Psychology: Human Perception and Performance*, 28(6), 1499–1501.
- Milner, T. E. (1992). A model for the generation of movements requiring endpoint precision. *Neuroscience*, 49(2), 487–496.
- Morasso, P., & Mussa Ivaldi, F. A. (1982, Sep). Trajectory formation and handwriting: A computational model. *Biological Cybernetics*, 45(2), 131–142.
- Muirhead, B. K. (2004). Mars rovers, past and future. In *Proceedings of the 2004 IEEE aerospace conference* (Vol. 1, pp. 128–134).
- Munir, S., & Book, W. J. (2002, Jun). Internet-based teleoperation using wave variables with prediction. *IEEE/ASME Transactions on Mechatronics*, 7(2), 124–133.
- Nelson, W. L. (1983). Physical principles for economies of skilled movements. *IEEE/ASME Transactions on Mechatronics*, 46, 135–147.
- Niemeyer, G., & Slotine, J.-J. (1997, Apr). Using wave variables for system analysis and robot control. In *Proceedings of the 1997 IEEE international conference on robotics and automation* (pp. 1619–1625). Albuquerque, NM.
- Niemeyer, G., & Slotine, J.-J. (2001). Toward bilateral internet teleoperation. In K. Goldberg & R. Siegwart (Eds.), *Beyond webcams* (pp. 193–213). Cambridge, MA: MIT Press.
- Nuño, E., & Basañez, L. (2005, Nov). Using force feedback for assisted robot teleoperation via ipv6 protocol in high-tech networks. In *36th international symposium on robotics*. Tokyo, Japan.
- Piazzi, A., & Visioli, A. (2000). Global minimum-jerk trajectory planning of robot manipulators. *Trans. on Industrial Electronics*, 47(1), 140–149.
- Rock, I., & Victor, J. (1964). Vision and touch: An experimentally created conflict between the two senses. *Science*, 143, 594–596.
- Schwartz, A. B., Moran, D. W., & Reina, G. A. (2004, Jan). Differential representation of perception and action in the frontal cortex. *Science*, 303, 380–384.
- Secco, E. L., & Magenes, A. V. G. (2004). Minimum jerk motion planning for a prosthetic finger. *Journal of Robotic Systems*, 21(7), 361–368.
- Shimojo, S., & Shams, L. (2001). Sensory modalities are not separate modalities: Plasticity and interactions. *Current Opinion in Neurobiology*, 11, 505–509.
- Smith, C., & Christensen, H. I. (2007, April). Using COTS to construct a high performance robot arm. In *Proceedings of the 2007 IEEE international conference on robotics and automation* (pp. 4056–4063). Rome, IT.
- Smith, O. J. M. (1959, Feb). A controller to overcome dead time. *ISA Journal*, 6, 28–33.
- Srinivasan, M., Beauregard, G. L., & Brock, D. L. (1996). The impact of visual information on the haptic perception of stiffness in virtual environments. In K. Danai (Ed.), *Proceedings of the 5th international symposium on haptic interfaces for virtual environment and teleoperator systems* (Vol. 58, pp. 555–559). Atlanta, GA, USA.
- Stein, B. E., & Meredith, M. A. (1993). *The merging of the senses*. MIT Press.
- Stein, M. R., & Paul, R. P. (1994). Operator interaction, for time-delayed teleoperation, with a behavior-based controller. In *Proceedings of the 1994 IEEE international conference on robotics and automation* (Vol. 1, pp. 231–236).
- Tsumaki, Y., Goshozono, T., Abe, K., Uchiyama, M., Koeppe, R., & Hirzinger, G. (2000). Experimental verification of an advanced space teleoperation system using internet. *Journal of Robotics and Mechatronics*, 12(4).
- Walker, M., & Orin, D. (1982). Efficient dynamic computer

simulation of robotic mechanisms. In *Transactions of the asme – journal of dynamic systems, measurement and control* (Vol. 104, pp. 205–211).

Welch, G., & Bishop, G. (2004, Apr). *An introduction to the kalman filter* (Tech. Rep. No. TR 95-041). Chapel Hill, NC: Department of Computer Science, University of North Carolina at Chapel Hill.

Whitney, D. E. (1987). Historical perspective and state of the art in robot force control. *Int. J. Rob. Res.*, 6(1), 3–14.



Christian Smith Received his M.Sc. in engineering physics from the Royal Institute of Technology, Stockholm, Sweden in 2005. He is currently a Ph.D. student at the Centre for Autonomous Systems, Royal Institute of Technology. Past work includes reinforcement learning for social robotics at ATR, Japan in 2004. Current research focuses on design and control of highly dynamic robotic systems inspired by studies of human nervous system.



Mattias Bratt Received his M.Sc. in engineering physics from the Royal Institute of Technology (KTH), Stockholm, Sweden in 1997. He has since been teaching introductory robotics at the same university, and is now a Ph.D. student at the Centre for Autonomous Systems, also at KTH. His current research deals with user interfaces for teleoperation, and the use of knowledge and ideas from the neuroscience field for improving such systems.



Henrik I Christensen received M.Sc. and Ph.D. degrees from Aalborg University, Denmark during 1987 and 1990, respectively. He is at present the KUKA Chair of Robotics at Georgia Institute of Technology and director of the Center for Robotics and Intelligent Machines. He has earlier served as the founding director of the Center for Autonomous Systems at the Royal Institute of Technology, and as a lecturer at Aalborg University. He has published more than 200 contributions across vision, robotics and artificial intelligence. Dr. Christensen

has served/is serving on a large number of editorial boards. In addition, he participates in a large number of research projects across 3 continents. He is a member of IEEE, ACM, AAAI and an officer of IFRR.